CMA EQUALIZATION OF MEASURED SOQPSK-TG DATA TRANSMITTED USING INET PACKET STRUCTURE

Itie Serge Kone Dossongui, Peter Taiwo

Faculty Advisors:
Dr. Arlene Cole-Rhodes and Dr. Farzad Moazzami

Department of Electrical and Computer Engineering
Morgan State University

ABSTRACT

In this work we consider the problem of recovering shaped offset quadrature-phase shift keying (SOQPSK)-TG modulated data, which has been transmitted over an unknown channel, using the iNET data packet structure. Previous work has shown the effectiveness of a block processing CMA equalizer which uses the known data bits contained in the iNET packet structure (i.e. the preamble and ASM bits) to provide an alternative method of initialization. In this research we apply a CMA equalizer, which has been initialized by the minimum mean square error (MMSE) equalizer to measure data that was transmitted using the iNET packet structure in a laboratory experiment. Since the CMA equalizer does not determine the correct phase shift for each data packet, the known data bits contained in each iNET packet will be used to determine the phase correction. The total number of bit errors will be used as a basis to evaluate the performance of our MMSE-initialized CMA equalizer for this experimental data.

1. INTRODUCTION

We investigate the effectiveness of the constant modulus algorithm (CMA) equalizer in recovering an unknown data bit stream, which has been transmitted using the iNET data packet structure over a channel determined by a laboratory environment. The modulation scheme used is the shaped offset quadrature-phase shift keying (SOQPSK), version TG. Initial test results on simulated data transmitted over the aeronautical channel have shown that CMA initialized by the MMSE equalizer provides a lower bit error rate ([3]) than the usual center-tap initialized CMA equalizer. When applied to actual measured data, CMA will sometimes introduce a phase shift of 180-degrees which can produce very large bit errors at the receiver if it is not corrected ([6]). In this paper we use the known data (i.e. the preamble and asynchronous marker (ASM)) bits in order to determine the additional phase shift introduced by the CMA equalizer. Thus we evaluate the performance of the CMA equalizer when the computed phase correction is used.

2. COMMUNICATION PACKET STRUCTURE

The data used in this experiment consist of 6336 bits as specified by the iNET packet structure. The iNET packet structure is made up of known data consisting of the preamble and asynchronous marker (ASM) bits, together with the actual (unknown) data. The length of the preamble is 128 bits; the length of the ASM is 64 bits, and actual data consists of 6144 bits. The preamble is known data, which is used to identify the start of the packet, and it is also used with the ASM to determine the phase shift introduced by the CMA after equalization. The iNET data packet structure is shown in figure 1.

Figure 1: INET data packet structure

At the transmitter, the data formatted is modulated using the SOQPSK-TG, and transmitted using the iNET packet structure over an unknown channel specified by a laboratory environment. For our experiments, the transmitted data is perfectly known, which allows us to determine and count the number of bit errors after applying the equalizer to the received data packets. We note that the received data packets are blocks of length N=6336 bits, consisting of at least 12672 samples since the modulator operates at an equivalent rate of two samples/bit.

3. CMA EQUALIZATION

At the receiver, the constant modulus algorithm (CMA) is used to specify a blind equalizer, which minimizes a specific cost function given by

$$J_{CMA}(y(n)) = E\left\{\left(|y(n)|^2 - R_2\right)^2\right\}, \tag{1}$$

where $R_2 = 1$ denotes the squared value of the CMA equalizer radius and this is usually calculated as $R_2 = E\{|c(i)|^4\} / E\{|c(i)|^2\}$ for a constellation type $c(i)$, and $y(n)$ represents the equalizer output block at time n. This cost function attempts to restore the shape of the signal by pulling the equalizer output onto a circle of radius R_2. Note that CMA does not take the signal phase into consideration.

The update of the equalizer weights is based on minimizing the CMA cost function. This is done using block processing, with a steepest descent rule given by

$$w_{p+1} = w_p - \mu \, \nabla_w J_{CMA}(y(n)) \ . \tag{2}$$

where μ_n is a step-size, $\nabla_w J_{CMA}(y(n))$ is the gradient of the cost function, and *y(n)* is the equalized output. Note that the MMSE equalizer is first computed based on the known preamble and ASM bits, and this is used as the initial weight vector, w_0 for the CMA update. The gradient vector of the cost function is given by

$$\nabla_w J_{CMA}(y(n)) = E\left\{4\left(\ |y(n)|^2 - R_2 \right) \ y(n) \ \underline{x}_n^{\ *}\right\} \tag{3}$$

where * operator denotes the complex conjugate operator. The output of the equalizer is a block of data determined by convolving the final equalizer weight vector, *w(n)* with the received data vector *x(n)*, i.e.

$$y(n) = \mathrm{w}(n) * x(n) \tag{4}$$

4. EXPERIMENTAL SYSTEM SET-UP

In this work, the performance of the CMA equalizer is evaluated on a data stream, which was produced experimentally at Brigham Young University (BYU) using the setup shown in Fig. 2. At this location the SOQPSK-TG transmitter was set to 10 Mbits/s using an internally generated length-2027 PN sequence, and the signal was transmitted over the air based on a channel which was determined by the walls and desks within their laboratory environment. The received signal was then filtered and down-converted to a 70 MHz IF where it was sampled at 100 Msamples/s. The sample sequence was stored on a hard drive using MATLAB® to create the low-pass equivalent signal at a sample rate equivalent to 2 samples/bit. This experiment was conducted by our collaborators at BYU, and they provided the data to us.

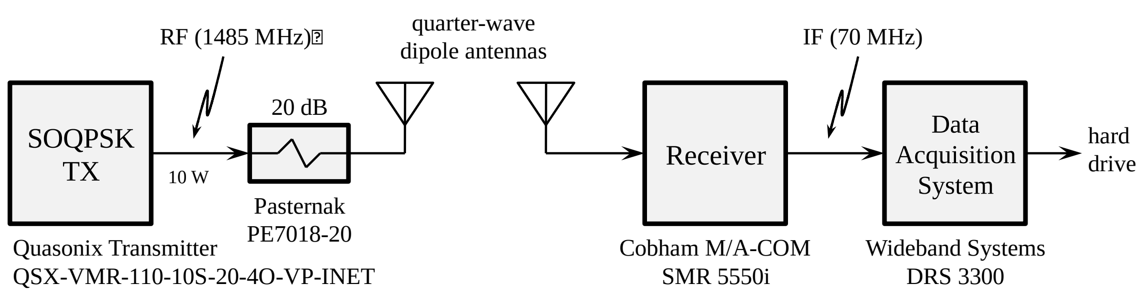

Figure. 2: Experimental setup used to generate the data used in the simulations

The CMA equalizer initialized by the minimum mean square error (MMSE) is applied to the received data as describe in ([6]). Figure 3 provides details on the processing and equalization of the received data.

Note that for this implementation of the CMA equalizer, the MMSE equalizer is computed using the known preamble and ASM bits within the first received packet, and that filter is used to initialize the CMA equalizer. This is then followed by an update of the filter coefficients based on a single iteration of the CMA equalizer. After this the equalizer coefficients continue to be updated once using the CMA update equation (2) for all subsequent received data packets. The detection filter is applied to the data, which is first down-sampled before the CMA phase correction is computed. Finally we count the errors by correlating the equalized data with the transmitted PN sequence of length 2047.

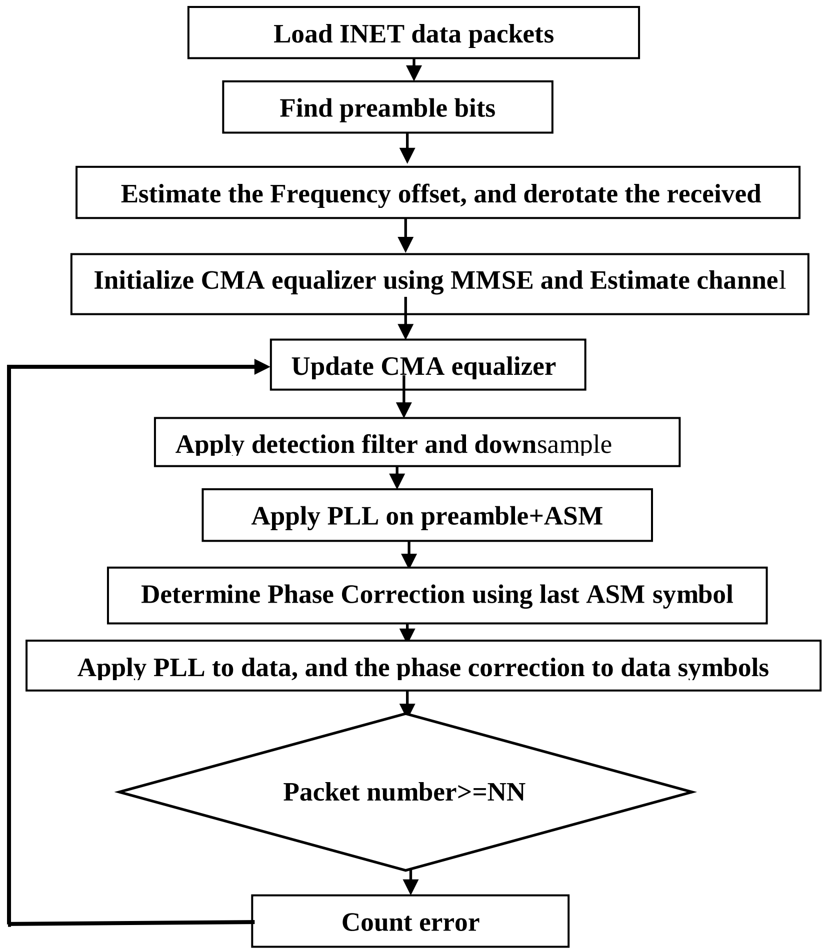

Figure 3: Receiver Data Processing

Procedure for Receiver data processing with Phase correction

- Step 1: Load received iNET data packets.
- Step 2: Find the Preamble bits.
- Step 3: Estimate the Frequency offset, and derotate the received samples.
- Step 4: For the first packet data, compute the MMSE equalizer filter and use this to initialize CMA.
- Step 5: Update the equalizer filter coefficients once using CMA, and apply to equalize current data packet.
- Step 6: Apply detection filter, downsample data in the packet, and apply phase lock loop (PLL) to the preamble and ASM bits.
- Step7: Use the last two ASM bits (i.e. last ASM symbol) to determine the CMA phase rotation of the current packet, then apply to correct the rotation of the data.

- Step 8: Repeat Steps 5 through 7 for each iNET packet at the receiver, until all packets are processed.
- Step 9: Determine the total number of bit errors by correlating the equalized data bits with the (known) transmitted PN sequence.

5. PERFORMANCE EVALUATION RESULTS

Initial tests on the experimental measured data have shown that the CMA introduces an additional phase shift, which cannot be corrected by the phase-lock loop (PLL). In this section we present the performance evaluation results for the MMSE-initialized CMA equalizer after the phase correction is determined and applied.

The phase correction was implemented as follows. The phase lock loop is applied to the entire received data packet, including the known preamble + ASM data bits. The CMA phase mismatch is determined by comparing the last ASM symbol in the iNET packet which is found before the transmitted data and is known. Assuming that the PLL has converged (and locked), the computed phase rotation for the last ASM symbol is applied to the unknown data after processing by PLL. The bit error rate is used as the basis for evaluating the performance of the MMSE-initialized CMA equalizer. Figures 4, 5 and 6 show the result of correlating the equalized data from the first 19 transmitted packets with the 2047 pn-sequence which was transmitted. The flipped correlations seen in figure 4 occur at the start of certain packets, and clearly indicates that some data packets are rotated 180 degrees out of phase with the input sequence. Thus corresponding bit errors are only shown in figures 5 and 6.

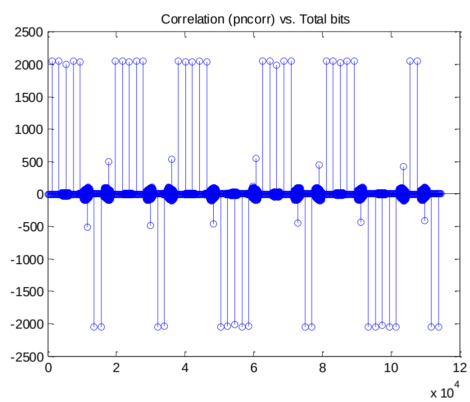

Figure 4: Correlation results for CMA equalizer (with no phase correction done)

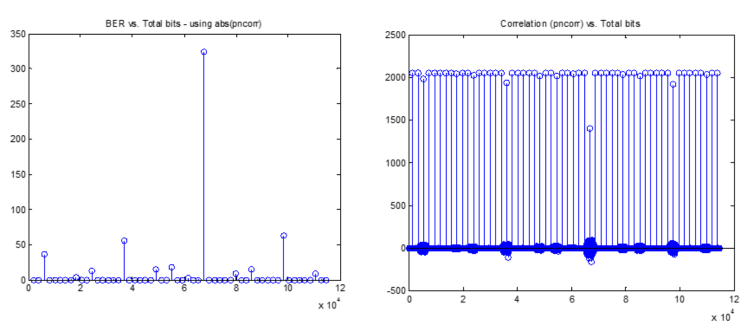

Figure 5: Error count results for CMA equalizer (phase correction has ASM in PLL)

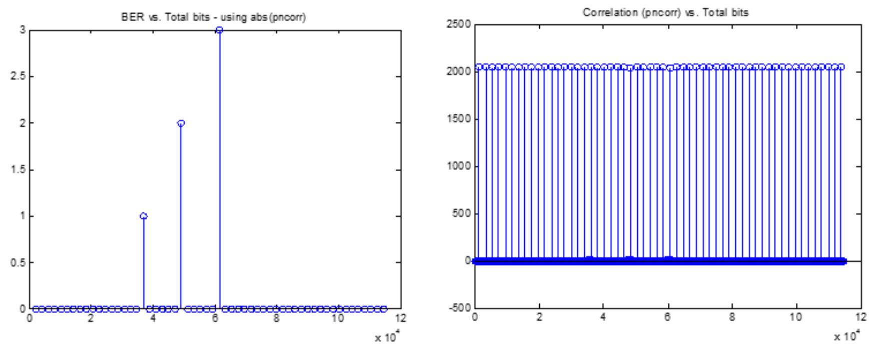

Figure 6: Error count results for CMA equalizer (phase correction has Preamble & ASM in PLL)

Figure 5 shows results of processing only the ASM and data through the PLL, while figure 6 shows the improvement when both the ASM and preamble are both processed before the data through the PLL. From figure 5, we note large error spikes occur, for example in packet 2 with 36 errors, packet 7 with 56 errors, packet 12 with 324 errors, and packet 17 with 63 errors so that the total number of errors for this case is 565 bits within 114632 transmitted bits.

	CMA + phase correction using Preamble and ASM	**CMA + phase correction using ASM only**	**MMSE equalizer**
No of Packets	19	19	19
Avg. peak separation	12673.05	12673.05	12673.05
Frequency offset (Hz)	-1578.7	-1578.7	-1012.8
BER	5.2341e-05	4.928816e-03	0.00e-00
No bit errors	6	565	0
Total Number of bits	114632	114632	114632

Table 1: Simulation results for CMA equalizer

Table 1 shows the simulation results for the MMSE equalizer, and the two CMA equalizers of figures 5 and 6 based on the first 19 packets. Note that the MMSE equalizer gives zero errors, while the total number of bit errors improved from 565 bits to 6 bits when the preamble is also included in the PLL.

Packet No.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
Phase shift of Equalized packet	0	0	π	0	0	π	0	0	π	π	0	π	π	0	0	π	0	0	π

Table 2: Computed CMA phase shift for the first 19 packets

Table 2 shows the computed CMA phase shift for each of the first 19 equalized data packets as determined based on the location of the last ASM symbol in each packet. Thus the results of figure 6 are produced by applying the phase corrections shown in Table 2 to the unknown data within each packet. By comparing the phase shifts of table 2 with the correlations shown in figure 4 we note that the phase rotations of packets 12 and 17 are computed incorrectly, and this may explain the corresponding large error spikes.

6. CONCLUSIONS

From this study, it is clear that in order for CMA to be successfully applied to equalize transmitted SOQPSK modulated data, there is additional processing which is necessary. In this case the iNET packet structure, which contains known bits at the start of the packet can be leveraged to allow for a phase correction providing the possibility that CMA can be used successfully as an equalizer for the transmitted SOQPSK data.

ACKNOWLEDGEMENTS

This work is being done in collaboration with Dr. Michael Rice of Brigham Young University. It is funded by the Test Resource Management Center (TRMC) Test and Evaluation Science and Technology (T&E/S&T) Program through the U.S. Army Program Executive Office for Simulation, Training and Instrumentation (PEO STRI) under contract W900KK_13-C-0026 (PAQ).

REFERENCES

[1] R. Johnson, et. al. "Blind Equalization using the Constant Modulus criterion: A Review", Proc, of IEEE, Vol.86, No.10, pp. 1927-1950, 1998.

[2] E. Law, "How well does a blind adaptive CMA equalizer work in a simulated telemetry multipath environment", International Telemetry Conference (ITC) 2004, San Diego, CA.

[3] A.Cole-Rhodes, H. Umuolo, F. Moazzami, M. Rice, "Real-Time CMA Equalization of SOQPSK for Aeronautical Telemetry", International Telemetry Conference (ITC) 2014, San Diego, CA.

[4] A.Cole-Rhodes, H. Umuolo, F. Moazzami, M. Rice, "A Block Processing Approach to CMA Equalization of SOQPSK for Aeronautical Telemetry", MILCOM 2014.

[5] M. Rice, "Phase 1 Report: Preamble Assisted Equalization for aeronautical telemetry (PAQ)," Brigham Young University, Technical Report, 2014, submitted to the Spectrum Efficient Technologies (SET) Office of the Science & Technology, Test & Evaluation (S&T/T&E) Program, Test Resource Management Center (TRMC). Also available on-line at http://hdl.lib.byu.edu/1877/3242.

[6] A.Cole-Rhodes, S. KoneDossongui, H. Umuolo, M. Rice, "Evaluating CMA equalization of SOQPSK-TG data for aeronautical telemetry", Proceedings of SPIE-STA, Baltimore 2015.